

**TUGAS PEKAN 6**  
**ALGORITMA DAN PEMROGRAMAN**  
**DO -WHILE LOOPS**

Disusun oleh:

Thaariq Salam

2511532022

Dosen Pengampu: Dr. Wahyudi S.T.M.T

Asisten Praktikum: Rahmad Dwirizki Olders



**DEPARTEMEN INFORMATIKA**

**FAKULTAS TEKNOLOGI INFORMASI**

**UNIVERSITAS ANDALAS**

**TAHUN 2025**

## KATA PENGANTAR

Puji syukur kehadirat Tuhan Yang Maha Esa atas rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan tugas praktikum mata kuliah Algoritma dan Pemrograman ini dengan baik dan tepat waktu. Tugas ini disusun untuk memenuhi salah satu syarat penilaian praktikum mengenai Tugas **Simulasi Game Lempar Dadu** dalam bahasa pemrograman Java.

Penugasan ini berfokus pada bertujuan untuk mendemonstrasikan pemahaman dan implementasi konsep-konsep dasar pemrograman berorientasi objek. Program yang dikembangkan berfokus pada aplikasi praktis dari struktur kontrol perulangan (*do-while*), percabangan bersyarat, serta penggunaan fungsi angka acak (*Random*) dan manajemen *input/output* dalam bahasa pemrograman Java.

Penulis menyadari bahwa makalah ini masih memiliki banyak kekurangan. Oleh karena itu, segala bentuk kritik dan saran yang konstruktif sangat kami harapkan demi penyempurnaan di masa mendatang.

Semoga makalah praktikum ini dapat memberikan manfaat, wawasan, dan menjadi panduan yang berguna bagi pembaca dalam memahami dasar-dasar logika pemrograman.

Padang, 5 November 2025

Penulis

## DAFTAR ISI

KATA PENGANTAR.....	i
DAFTAR ISI .....	ii
<u>A.</u> Deskripsi Program .....	1
<u>B.</u> Pseudocode .....	2
<u>1.</u> Judul :.....	2
<u>2.</u> Deklarasi:.....	2
<u>3.</u> Pseudocode .....	3
<u>C.</u> Flowchart .....	4
<u>D.</u> Kode Program.....	5
<u>E.</u> Output Program.....	6
F. Kesimpulan.....	6

## A. Deskripsi Program

Program Java ini dirancang sebagai implementasi praktis dari konsep Angka Acak, Struktur Kontrol Perulangan (do-while), dan Percabangan Bersyarat (if-else) serta Input/Output dasar. Tujuan utama program ini adalah menyajikan simulasi interaktif pelemparan dua buah dadu hingga kondisi kemenangan (total 7) tercapai atau pengguna memilih untuk berhenti.

### 1. Tujuan dan Konsep Utama

- Tujuan:

Mensimulasikan permainan lempar dua dadu secara berulang dengan tujuan utama mencapai total penjumlahan mata dadu sama dengan 7.

- Konsep yang Diimplementasikan:

- Angka Acak: Memanfaatkan kelas `java.util.Random` untuk menghasilkan nilai mata dadu secara *fair* dan acak (rentang 1 hingga 6).
- Perulangan: Menggunakan struktur *do-while* untuk mengulang proses lemparan dadu. *Loop* utama berlanjut selama pengguna belum menang dan memilih untuk melempar lagi.<sup>2</sup>

### 2. Komponen dan Struktur Program

Program ini dibangun menggunakan struktur dan komponen berikut:

- Kelas *Random*: Digunakan untuk menghasilkan dua bilangan bulat acak (dadu 1 dan dadu 2) pada setiap iterasi.
- Kelas *Scanner*: Digunakan untuk menerima input dari pengguna, yaitu konfirmasi apakah pengguna ingin melempar dadu lagi ("ya" atau "tidak").
- Variabel Bertipe *int*: Digunakan untuk menyimpan hasil lemparan (dadu1, dadu2, total) dan menghitung jumlah upaya (jumlahPercobaan).
- Variabel Bertipe *boolean*: Variabel lemparLagi berfungsi sebagai flag untuk mengontrol kondisi keluar dari loop utama.
- Struktur Kontrol Perulangan (*do-while*):
  - Perulangan Luar: Mengontrol keseluruhan permainan; berjalan minimal sekali dan terus berlanjut hingga kondisi kemenangan tercapai (total == 7) atau pengguna memilih "tidak".

- Perulangan Dalam (Nested Loop): Digunakan untuk memvalidasi input pengguna, memastikan pengguna hanya memasukkan "ya" atau "tidak".3.
3. Logika Kemenangan dan Terminasi
- Kondisi Kemenangan:  
Program mencatat kemenangan jika dadu1 + dadu2 = 7. Setelah menang, program menampilkan jumlah percobaan yang dibutuhkan dan menghentikan perulangan.
  - Kondisi Gagal/Berhenti:  
Jika hasil lemparan bukan 7, program menawarkan opsi untuk melanjutkan. Jika pengguna memilih "tidak", program menampilkan pesan "Anda gagal menang" dan menghentikan perulangan.
  - Alur Sekuensial:  
Di setiap iterasi, program berjalan secara sekuensial: lempar dadu > hitung total > tampilkan hasil > cek kondisi menang > tanyakan pengguna..

## B. Pseudocode

1. Judul :

Game Lempar 2 Dadu hingga jumlahnya 7

2. Deklarasi:

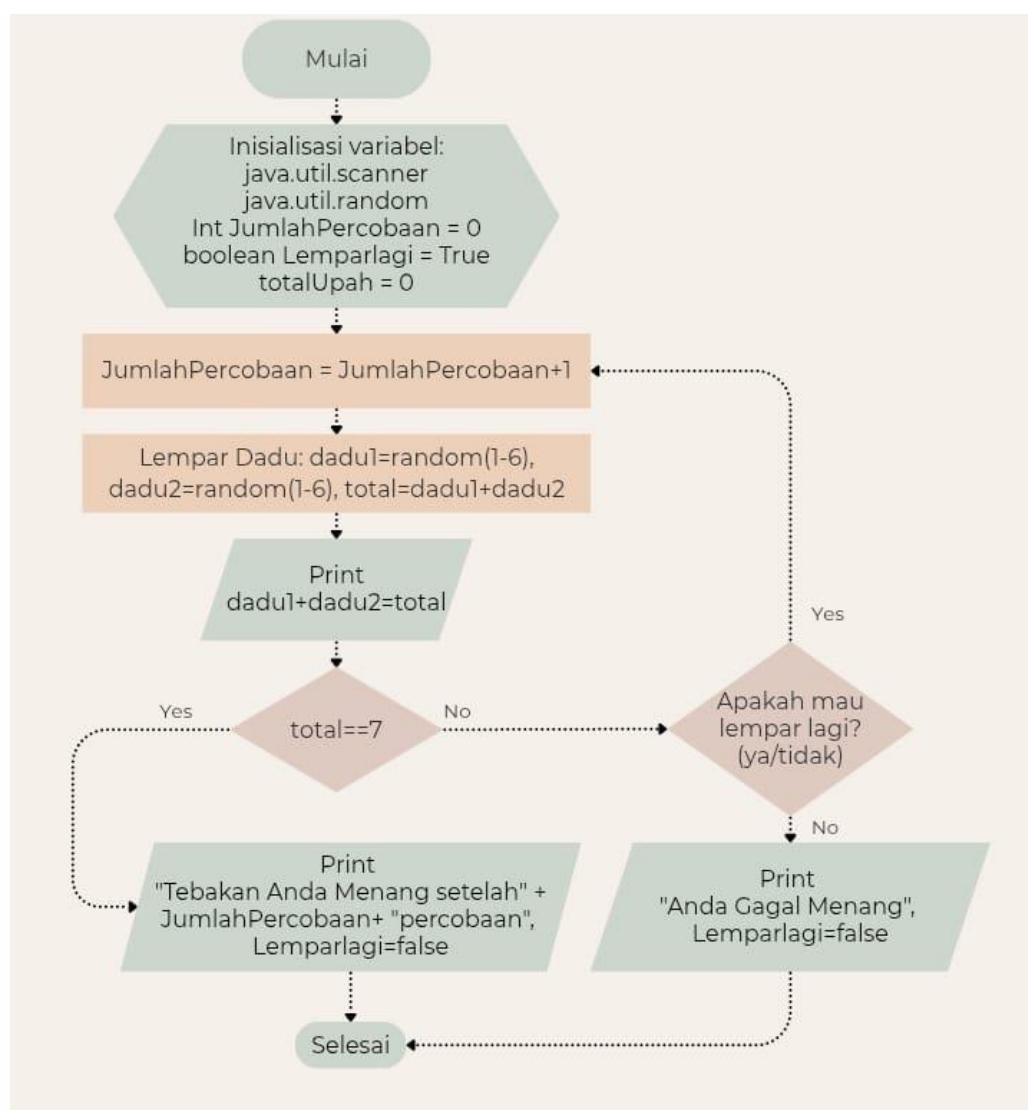
1. Deklarasikan Random = Objek\_Random
2. Deklarasikan Scanner = Objek\_Scanner
3. Deklarasikan Integer Jumlahpercobaan = 0
4. Deklarasikan Boolean Lemparlagi = True

### 3. Pseudocode

```
1. // Mulai Loop Utama (Do-While)
2. Ulangi
3. Jumlahpercobaan = Jumlahpercobaan + 1
4. // Lempar Dadu Secara Acak (1 Sampai 6)
5. Dadu1 = Generate_Random_Int(1, 6)
6. Dadu2 = Generate_Random_Int(1, 6)
7. Total = Dadu1 + Dadu2
8. // Tampilkan Hasil Lemparan
9. PRINT Dadu1 + " + " + Dadu2 + " = " + Total
10. IF Total == 7 THEN
11. PRINT "Tebakan Anda Benar"
12. PRINT "Anda menang setelah " + Jumlahpercobaan + " Percobaan."
13. Lemparlagi = False // Hentikan Loop
14. ELSE
15. // Mulai Loop Untuk Pilihan (Do-While True)
16. Ulangi
17. PRINT "Apakah Anda Mau Lempar Dadu Lagi? (Ya/Tidak): "
18. READ Pilihan
19. IF Pilihan Sama Dengan "Tidak" THEN
20. PRINT "Anda Gagal Menang"
21. Lemparlagi = False // Hentikan Loop Utama
22. Hentikan Loop Saat Ini // Keluar Dari Loop Pilihan
23. IF Pilihan Sama Dengan "Ya" THEN
24. STOP Loop Saat Ini // Keluar Dari Loop Pilihan, Lanjut Loop Utama
25. ELSE
26. PRINT "Input Tidak Valid. Mohon Masukkan 'Ya' Atau 'Tidak'."
27. END IF
28. Sampai Ketika False // Loop Pilihan Berjalan Terus Sampai Hentikan Loop
   Saat Ini Dipanggil
29. END IF
```

30. Sampai Ketika Lemparlagi Sama Dengan False // Loop Utama Berlanjut Selama Lemparlagi Adalah True
31. CLOSE Scanner
32. Selesai

### C. Flowchart



Gambar 1 : Flowchart

#### D. Kode Program

```
1. package pekan6_2511532022;
2. import java.util.Random;
3. import java.util.Scanner;
4.
5. public class TugasPekan6_2511532022 {
6.
7.     public static void main(String[] args) {
8.         Random random = new Random();
9.         Scanner scanner = new Scanner(System.in);
10.
11.         int jumlahPercobaan = 0;
12.         boolean lemparLagi = true;
13.
14.         do {
15.             jumlahPercobaan++;
16.             int dadu1 = random.nextInt(6) + 1;
17.             int dadu2 = random.nextInt(6) + 1;
18.             int total = dadu1 + dadu2;
19.
20.             System.out.println(dadu1 + " + " +
dadu2 + " = " + total);
21.
22.             if (total == 7) {
23.                 System.out.println("Tebakan Anda
Benar");
24.                 System.out.println("Anda menang
setelah " + jumlahPercobaan + " percobaan.");
25.                 lemparLagi = false;
26.             }
27.             else {
28.                 String pilihan;
29.                 do {
30.
System.out.println("Tebakan Anda Salah");
31.                 System.out.print("Apakah
Anda mau lempar dadu lagi? (ya/tidak): ");
32.                 pilihan =
scanner.nextLine().toLowerCase();
33.
34.                 if
(pilihan.equals("tidak")) {
35.
System.out.println("Anda gagal menang");
36.                 lemparLagi = false;
37.                 break;
```

```

38.                     } else if
39.             (pilihan.equals("ya")) {
40.                 break;
41.             } else {
42.                 System.out.println("Input tidak valid. Mohon masukkan
43. 'ya' atau 'tidak'.");
44.             }
45.         }
46.
47.     } while (lemparLagi);
48.
49.     scanner.close();
50. }
51. }
```

## E. Output Program

```

6 + 6 = 12
Tebakan Anda Salah
Apakah Anda mau lempar dadu lagi? (ya/tidak): ya
1 + 3 = 4
Tebakan Anda Salah
Apakah Anda mau lempar dadu lagi? (ya/tidak): tidak
Anda gagal menang
```

Gambar 2 : Output Program 1

```

6 + 3 = 9
Tebakan Anda Salah
Apakah Anda mau lempar dadu lagi? (ya/tidak): ya
1 + 6 = 7
Tebakan Anda Benar
Anda menang setelah 2 percobaan.
```

Gambar 3 : Output Program 2

## F. Kesimpulan

Program simulasi game lempar dadu ini berhasil mengimplementasikan penggunaan struktur kontrol esensial dalam Java, yaitu perulangan (*do-while*) dan

percabangan (*if-else*), yang didukung oleh fungsi pembangkitan angka acak (*java.util.Random*).

Poin-Poin Utama:

1. Fungsi dan Mekanisme: Program secara efektif menyajikan simulasi interaktif di mana pengguna berulang kali melempar dua dadu hingga tercapai total 7 (kondisi menang) atau pengguna secara eksplisit memilih untuk berhenti (kondisi gagal/berhenti).
2. Struktur Kontrol: Penggunaan loop do-while memastikan permainan berjalan secara berkelanjutan dan memungkinkan perhitungan akurat terhadap jumlah percobaan yang dibutuhkan untuk mencapai kemenangan.

Secara keseluruhan, program ini tidak hanya memenuhi tujuan untuk mensimulasikan permainan dadu sederhana, tetapi juga berfungsi sebagai demonstrasi yang jelas mengenai penerapan fundamental pemrograman Java dalam membangun aplikasi berbasis interaksi dan logika perulangan.